原　著　論　文

敵対生成ネットワークによる文書分類

岩田一樹
東北福祉大学

要旨

本研究では、小論文やレポートの自動評価を目標に、教師なし学習の一つである敵対生成ネットワークの Discriminator に着目した 1 文の分類をタスクに実験を行った。敵対生成ネットワークには SeqGAN を用い、訓練データには夏目漱石の著書の文を用いた。その訓練データによる学習後、学習済みの Discriminator に、訓練データとは別の夏目漱石の著書における文、および、太宰治・芥川龍之介の著書の文を与え、夏目漱石の文の識別率を計算した。また、いくつかの教師あり学習モデルで同じタスクの学習を行い、結果の比較を行った。その結果、敵対生成により得られた Discriminator の識別性能が最も劣っており、その要因の一つには、Generator の生成する文が未熟であったことがあげられた。

キーワード : 敵対生成ネットワーク、文書分類、深層学習

緒言

敵対生成ネットワーク（Generative Adversarial Network：GAN）は、2014年に I. Goodfellow らが考案した教師なし学習の生成モデルである[1)]。この生成モデルは、訓練データを与え、その訓練データの特徴を反映した類似のデータを機械に生成させるものである。当初、GAN は画像データを対象として提案され、それまで不可能だった、機械による鮮明な画像の自動生成に成功し、大きな注目を集めた。そして、現在において、機械学習分野において最も盛んに研究されているアルゴリズムの 1 つである。

GAN は、生成器である Generator と識別機である Discriminator という 2 つの機構を有し、それら 2 つを敵対させ、競わせながら学習を進める（Fig. 1）。具体的に、Generator は与えられた訓練データを基にそれらの特徴を反映した類似のデータを目指してデータ生成を行うように学習する。それに対して、Discriminator には Generator が生成したデータと訓練データが与えられ、与えられたデータそれぞれが Generator によって生成されたものなのか、訓練データなのか、を正確に鑑別できるように学習する。この敵対関係にある 2 つの機構を競わせながら学習を行った結果、最終的に、Generator は訓練データの特徴を反映したデータを生成できるようになる。一方、Discriminator データの識別を通して、Generator の生成データがどのくらい訓練データの特徴を反映しているかを評価できるようになる。つまり、Discriminator は入力されたデータが訓練データか、生成データかを確率として出力するが、その確率が訓練データにどのくらい近いかの指標となるということである。

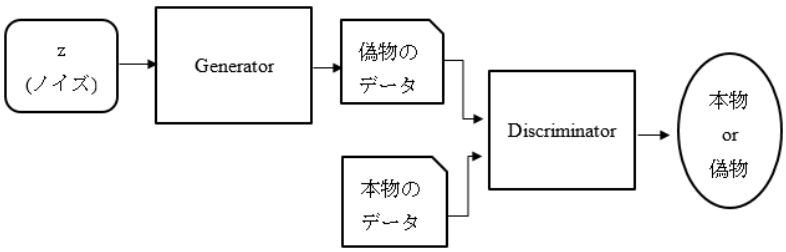

Fig. 1　GAN の概念図

GAN は、訓練データのラベルを 1 、Generator の生成画像のラベルを 0 とした際に、以下の損失関数 V について、Discriminator である D に対しては最大化、Generator である G に対しては最小化することを目指して、学習を行う[1)]。

$$V(G,D) = \mathbb{E}_{x\sim p_{t(x)}}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]. \cdots (1)$$

ここで、x は訓練データ、G(z) は生成データを示している。また、𝔼 は期待値を意味しており、$x\sim p_{t(x)}$ は訓練データからの抽出、$z\sim p_z(z)$ は Generator が生成のシードとする z 空間のノイズからの抽出を意味する。

まず、Generator を固定して考えると、右辺第 1 項において、D(x) は入力が訓練データなので全て 1 を出力するのが理想であり、一方、G(z) は生成データなので D(G(z)) は全て 0 を出力するのが理想である。その結果、理想的な Discriminator において上式の V は最大値の 0 となる。（ラベルが 0 と 1 しかなく、その対数を取るので、V は正の値になることはありえない。）それに対し、Discriminator を固定して考えると、上式の右辺第 2 項の D(G(z)) が全て 1 であるときに Discriminator を完全に騙せている理想的な Generator となるので、第 1 項に関わらず、V は最小値、－∞になる。実際には、理想的とはなりえないので、有限の最小値となる。したがって、V を最大にする D と最小にする G の関数を決めるのが GAN の学習といえる。その後、このモデルは画像のみでなく、文章や音楽などの系列データにも適用されるようになり、現在、最も盛んに研究されている生成モデルとなっている。

また、画像を扱う場合には、生成器、ならびに、識別器のネットワークには畳み込みニューラル・ネットワーク（Convolutional Neural Network；CNN）が用いられることが多い[2)]。CNN は、位置など局在的な変化に堅牢な局在不変性を仮定可能なニューラル・ネットワークのモデルで、最近の機械学習分野において中心的なアルゴリズムの 1 つであり[3)]、近年では、自然言語処理においても、CNN が利用されている[4)]。

一方、近年、大学教育において、アクティブ・ラーニングが重要視されるようになり、小論文やレポートを評価課題とすることが多くなった。そして、それらを評価するにあたって、採点者の時間的負担の軽減、および、評価における採点のゆらぎの低減を目的に、自動評価システムの研究が行われるようになった[5)]。

従来の自動評価システムは、提出された文章を入力とし、その評価結果の出力をタスクとしている。そして、これらシステムの構築は、既存の「文章」と人が行ったその文章の「評価」の組を訓練データとして与え、その訓練データを元に機械が学習し、非訓練データ（この場合は、評価対象のレポート）の評価に適用する、または、採点においてヒューリスティックに重要と判断した量を特徴量として、それを基に評価する、または、これら 2 つの評価を組み合わせるといったアプローチが取られている。

上記の様に、訓練データを「入力データ」とその「正解」の組で与える学習は「教師あり学習」と呼ばれ、機械学習において重要であり、かつ、最も成功した1分野を築いている。しかし、教師あり学習においては、訓練データを作成する際に①大量の文章を人力によって評価する必要がある点、②人力で評価を行うため、訓練データの「正解」に評価ゆらぎが包含される可能性、が問題となる。特に、学習に数十万～数百万、または、それ以上の数の訓練データを学習に要するようになった今日、①の問題が大きくなっている。一方、ヒューリスティックに特徴量を設定する場合、その根拠が経験的なものであるため、その特徴が正しく評価の基準を捉えているかを確認するのが困難であると同時に、経験の共有が難しいために定義が困難な論理性の評価は不可能といえるのが現状である。

一方、訓練データに「正解」を与えない機械学習のことを「教師なし学習」と呼び、教師なし学習においては、訓練データは「入力データ」のみで、その「正解」にあたるものを有さない。したがって、訓練データに対する人力による評価が不要であるため、教師あり学習よりも大きな訓練データを比較的容易に取得することが可能となり、上記①の問題を克服できる。また、人が評価を行わないので、②の問題はそもそも生じない。したがって、教師なし学習であるGANを利用することによって、上記①と②の課題を解決し、また同時に、従来と異なる機械のみの判断をベースにした小論文やレポートの自動評価が実現する可能性がある。そのためには、文章とは文の連なりであることから、1文がそのテーマの文書らしいか、そうでないかの識別を行う必要がある。1文の評価が可能になれば、例えば、文書全体で何文がそれらしいかで全体の評価が可能となる。

本研究では、GANをベースとした小論文やレポートの自動評価の実現を目指し、それによる1文の識別が可能かを検討するために、GANによってトレーニングされたDiscriminatorを用いた文書分類を行った。具体的には、夏目漱石の文を学習データに用い、GANにより学習させた後に、学習データに用いた以外の夏目漱石の文、太宰治の文、芥川龍之介の文の3種類を用いて、夏目漱石の文とそれ以外を分離できるのか実験を行った。この実験の意図は、レポートの評価を“レポートらしさ”で行うことを想定し、レポートの文中でそれらしいと識別される文の割合が反映すると推察しているからである。つまり、夏目漱石らしい文章を、文中で夏目漱石の文と識別される文の数で評価するために、1文が夏目漱石らしいかを機械が識別できるかを検討するということである。

また、「教師あり学習」の識別機との性能比較を行うために、代表的な識別モデルであるLong term Short Memory（LSTM）構造、CNN構造、TextCNN構造の識別器も作成し検討した。なお、GANのDiscriminatorもTextCNNベースのものを用いる。

研究方法

本節では、分散表現、本研究で用いるネットワーク構造の説明を行った後に、SeqGANについて述べる。そして、その後、データセットと評価指標について述べる。

分散表現

ニューラル・ネットワーク・ベースに限らず、自然言語処理分野において、機械は言語についての内部表現を有さないので、文書は単語毎にIDが振られてはじめて機械の分析対象となる。IDが振られた後、単語をベクトルとして扱うことになるが、ニューラル・ネットワーク・ベースなどではone hotベクトルと呼ばれる、次元を登場単語数、そして、自らのIDを要素番号とする要素のみが1のベクトルに変換して扱われる。このone hotベクトルは日本語の文書群では10,000次元程度を有し、1要素のみが1のスパースと呼ばれる非常に疎なベクトルとなる。しかし、一般的にスパースにおいて、ほとんどのベクトルの要素が0であるため、興味があり必要とされる情報は僅かにしか含まれず、次元を減らす次元圧縮を行う。

自然言語処理において、この次元圧縮することを分散表現と呼び、10,000程度の次元を100～300程度に次元圧縮する。分散表現の手法には主成分分析やトピックモデル[6]などいくつかの手法があるが、ニューラル・ネットワーク・ベースの処理においては、その親和性から word 2 vec 表現[7]を用いる場合が多い。本研究においても分散表現には、word 2 vec 表現を用い、圧縮後の次元は100次元とした。

Long Short-Term Memory（LSTM）

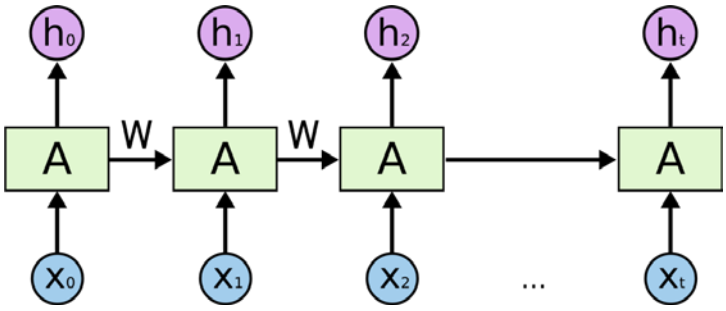

Fig. 2　シンプルな RNN の内部構造

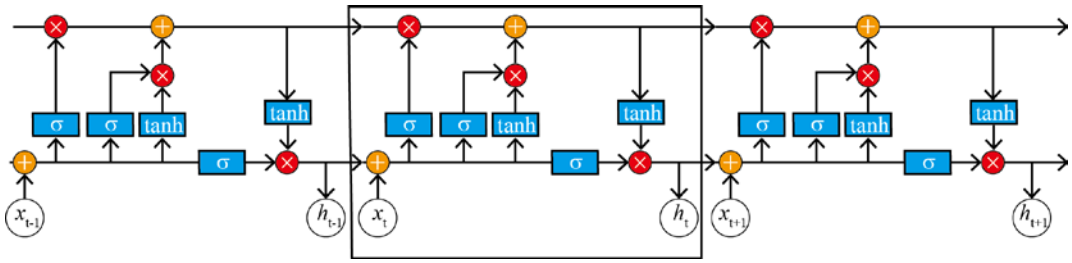

Fig. 3　LSTM の内部構造

LSTM の説明の前に、系列データを扱うニューラル・ネットワークの１つである再帰的ニューラル・ネットワーク（Recurrent Neural Network；RNN）について述べる[8]。先ず、RNN の内部構造を Fig. 2 に示す。Fig. 2 において、X=（$\vec{x_0}$，$\vec{x_1}$, $\vec{x_2}$ … $\vec{x_t}$）は系列を有するベクトルの入力データで、0 → 1 → 2 … → t の順番を有する。本研究においては、分散表現された単語のベクトルがこれに該当する。また、A はニューラル・ネットワーク、H=（h_0，h_1，h_2 …h_t）は入力 $\vec{x_i}$（$0 \leq i \leq t$）に出力値（h_i）、W は系列間の重みを意味する。なお、図２は概念図であるため、系列間のバイアス項は省略している。そして、t 番目の出力 h_t を、t 番目の入力である x_t による出力と t-1 番目の出力に由来する Wh_{t-1}の和として下式で与える。

$$h_t = AX_t + Wh_{t-1}.$$

この計算を逐次的に行うことで系列を取り扱うモデルが RNN である。

ただ、この RNN モデルには W の積とみなされる項があるため、W が１付近でない場合、系列情報が０になってしまったり（W ＜ １）、逆に、発散してしまったり（W ＞１）する問題があった。この問題を「忘却ゲート」「入力ゲート」「出力ゲート」と呼ばれる３つの機構を用いて改善したモデルが Long Short-Term Memory（LSTM）と呼ばれるものである[9]。LSTM の概念図を Fig. 3 に示す。ここで、σ はシグモイド関数、tanh は双曲線正接・余接関数を意味する。

枠で囲まれた部分が t=t における LSTM の処理の部分であり、単純な RNN との大きな違いは、前の枠

(t-1) から 2 つの情報（2 つの矢印）を引き継ぎ、同じく、t+ 1 に 2 つの情報を渡す点である。大まかに言って、RNN では下方の矢印のみである。ゲートとは基本的にシグモイド関数の部分で、シグモイド関数が（0, 1）の出力であるため、それとの積（×）の箇所においては、引数に対してシグモイド関数の出力が 0 に近いときには不要な情報として継承を遮断し、1 に近い時は必要な情報として引き継ぐことになる。これによって、人における忘却と記憶を表現している。なお、Fig. 3 で σ がある箇所が左から忘却ゲート、入力ゲート、出力ゲートである。また、双曲線正接・余接関数は（-1, 1）を出力する関数なので、定性的には、引数となる情報に + か − の符号を付けているとみなされる。

この LSTM は、文書を単語の系列データとみなして文書分類のタスクに用いられることもある一方で、t-1までの情報を元に t を予測するタスクもこなせるモデルであるため、識別のタスクにも生成のタスクにも利用される[10]。本研究においても、この LSTM を文書識別、および、GAN おいて Generator として用いている。

識別タスクにおける LSTM の構造は one hot ベクトルを100次元に圧縮する分散表現層、LSTM 層、出力層とし、入力→分散表現層（100）→ LSTM 層（セルサイズ：256）→出力層（2）とした。最後の出力層の出力が 2 なのは、タスクが夏目漱石の文か、そうでないかの 2 値分類タスクだからである。また、学習は、オプティマイザに RMSprop を学習係数0.0001で用い、学習回数は500 epochs とした。

Convolutional Neural Network（CNN）

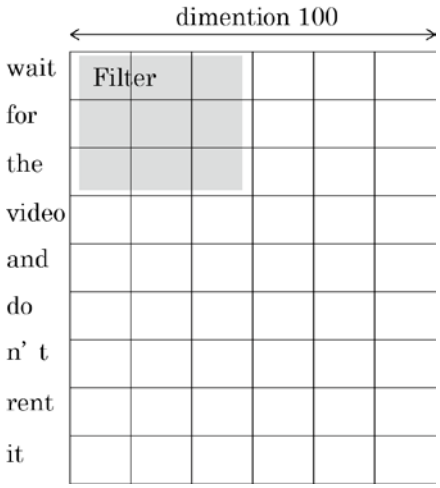

Fig. 4　CNN の概念図

畳み込みニューラル・ネットワーク（CNN）は、位置など局在的な変化に堅牢な局在不変性を仮定可能なニューラル・ネットワークのモデルで、近年の機械学習分野、特に、深層学習分野（Deep Learning）において中心的な役割を果たしているネットワーク構造である。通常、文書において CNN を用いる場合は、次に述べる TextCNN を用いるが、本研究では実験的に、分散表現した文書を Fig. 4 で示した 2 次元配列（画像）として捉え、画像を処理する際の手法でも文書分類を行った。

1 文の最大長は20単語、分散表現で100次元とし、1 文を20 pix、100 pix の画像と等価に扱い、CNN 構造は全ての畳み込み層のフィルタのカーネルサイズを 3 x 3 、ストライドを 1 とし、入力層→分散表現層→ 2 次元配列化（20x100）→畳み込み層（フィルタ数：16）→畳み込み層（フィルタ数：16）→畳み込み層（フィルタ数：32）→畳み込み層（フィルタ数：32）→全結合層（ノード数：512）→出力層（ノード数：512）→出力（2）である。なお、本研究では pooling 層を用いていない。また、学習は、オプティマイザに RMSprop を学習係数0.001で用い、学習回数は500 epochs とした。

TextCNN

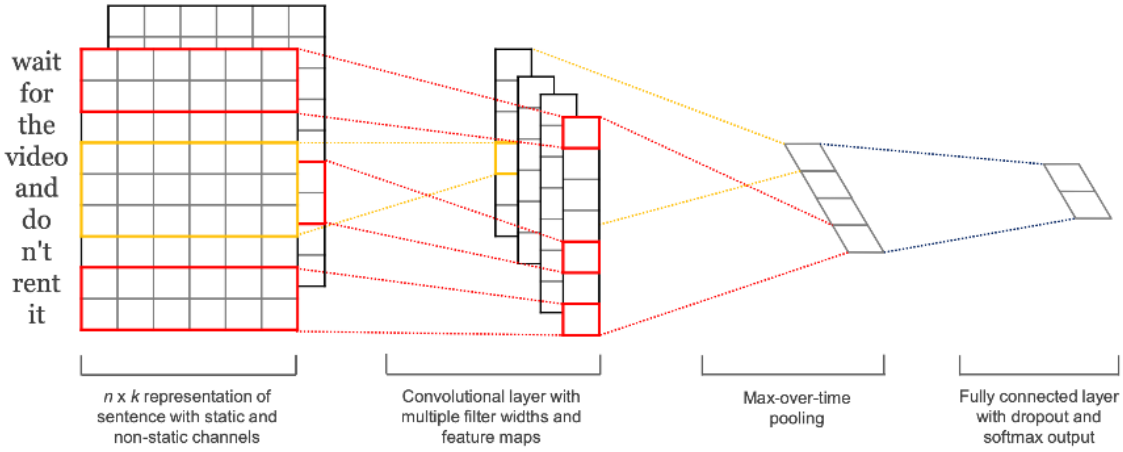

Fig. 5　TextCNN の概念図（5）から引用

TextCNN は CNN を使ってテキスト分類に取り込んだ手法で、文分類を目的としている[4)]。したがって、長い文書の分類よりは、1 文まだは複数の文からなる短い文書の分類に適しており、感情分析、評判分析や質問タイプの分類などの短い文書を対象とするタスクに用いられる。

TextCNN のネットワーク構造の概念図を Fig. 5 に示す。左から右へ、入力から出力となっていて、左から、分散表現層、畳み込み層、Max プーリング、全結合層、出力層となっている。上記の CNN との違いは、上記 CNN が分散表現された 1 文を“画像”として取り扱い、一般的な CNN 構造で処理しているのに対して、TextCNN は畳み込みに用いるフィルタが、フィルタに入れる単語数×分散表現のベクトル次元となっている点と、Max プーリング層において、1 文全体から最大値を取得し、それを最後の全結合層への 1 入力としている点である。

CNN と同じく、文の最大長は20として、TextCNN 構造は、入力層→分散表現層→ 2 次元配列化（20x100）→畳み込み層→ Max Pooling 層→出力層（2）である。なお、畳み込み層において、1 フィルタに入る連続する単語数は [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20] で、それぞれに対応するフィルタ数は [100, 200, 200, 200, 200, 100, 100, 100, 100, 100, 160, 160] である。具体的には、連続する 2 単語のフィルタは 2 x100のフィルタ、3 単語のフィルタは 3 x100などのカーネルサイズとなる。そして、その 1 つの畳み込みの結果全体で Max Pooling を取るので、Max Pooling 層の出力は1,720となり、それから出力層の 2 を得る構造になっている。学習は、オプティマイザに RMSprop を学習係数0.0001で用い、学習回数は500 epochs とした。

なお、この構造は次項で説明する、SeqGAN における Discriminator と同じである。

SeqGAN

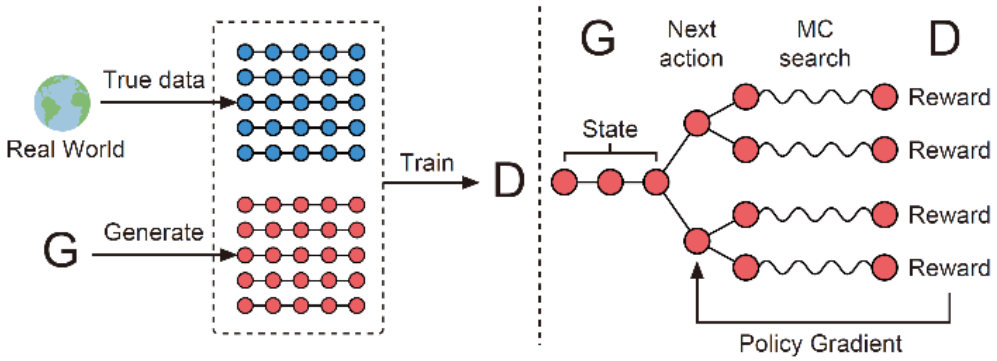

Fig. 6　SeqGAN の概念図（6）から引用

本研究で検討すべき Discriminator を得るための GAN には、文章生成を目的としている SeqGAN を用いた[10), 11)]。SeqGAN は、系列を有さない画像のようなデータを対象とした GAN と異なり、文のような系列（順番がある）データを扱うモデルである。そのため、Generator には系列データの取り扱いが可能な上記で説明した LSTM を先行研究から用い、Discriminator には TextCNN を用いた。また、教師あり学習時の TextCNN との比較ができるよう、TextCNN の構造は前節の構造で統一している。

SeqGAN の特徴は、系列データを生成するにあたって、1 文字目から t-1文字目までの単語系列：$Y_{1:t-1}$ から t 番目の単語 y_t を予測するために、Fig. 6 の概念図にあるように y_{t+1} 以降をモンテカルロ法（乱数）により規定回数回生成し、その結果も含めて真偽を識別器に判定させ、それらを元に、$Y_{1:t-1}$ から y_t を生成した結果の評価値を決定する点である。このように処理をする理由は、順番を有さない、例えば、画像データであれば、単に Generator の出力に対して、Discriminator が評価を行えばよいが、系列がある場合、Generator は文全体を 1 回に生成するのではなく、1 単語目から 1 単語ずつ順番に生成していくためである。一方、Discriminator は 1 文全体に対して真贋の判定をするため、生成中途中を扱うことが出来ないためである。また、別の見方をすると、敵対学習によって、LSTM は t 文字目を生成の適切さ学習し、モンテカルロ法と Discriminator によって、文全体の文法などの整合性を学習しているとも解釈できる。

SeqGAN における生成器の誤差関数（J_θ）は下式によって与えられる。

$$J_\theta = \sum G_\theta(y_t|Y_{1:t-1}) Q_{D_\phi}^{G_\theta}\left(Y_{1:t-1}, y_t\right).$$

$$Q_{D_\phi}^{G_\theta}(s = Y_{1:t-1}, a = y_t) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\phi(Y_{1:T}^n),\ Y_{1:T}^n \in \mathrm{MC}^{G_\beta}(Y_{1:t}; N) & \text{for } t < T. \\ D_\phi(Y_{1:t}) \text{ for } t = T. \end{cases}$$

ここで、G_θ（G_β）は θ（β）をパラメータとする LSTM を用いた生成器、D_ϕ は ϕ をパラメータとする TextCNN を用いた識別器、Q は報酬関数である。この誤差関数は、$Y_{1:t-1}$ から y_t を生成する尤度を報酬関数で表し、それを生成器によって $Y_{1:t-1}$ から y_t を生成される確率で期待値をとることに相当する。また、報酬関数は、$Y_{1:t-1}$ から y_t を生成器によって生成した後、以降をパラメータ更新する前の生成器（G_β）からモンテカルロ法とマルコフ連鎖により複数生成し、その生成結果を識別器で評価、その上で、平均化したものである。この誤差関数によって、生成器は報酬関数が最大となるようにパラメータ θ を方策勾配法にて更新する。

一方、識別器は一般のGANと同様に、

$$\mathrm{V} = \mathbb{E}_{\mathrm{Y}\sim\mathrm{data}}\left[\log D_{\phi}(\mathrm{Y})\right] + \mathbb{E}_{\mathrm{Y}\sim G_{\theta}}\left[\log\left(1 - D_{\phi}(\mathrm{Y})\right)\right].$$

が最大になるようϕを更新する。なお、この式は識別器が全ての入力に対して真偽を正しく識別した際に最大値0となる。

本研究におけるSeqGANのGeneratorは6）と同じネットワーク構造とし、Discriminatorは TextCNNの説明したものと同じ構造を用いた。また、学習は10）の文献を参照し、Generatorのプレトーニングを200回、Discriminatorのプレトレーニングを5回、敵対学習は500回行った。また、Generator、Discriminatorともにオプティマイザには RMSpropを用い、学習率はそれぞれ、0.01と0.0001とした。そして、モンテカルロ法での分生成の回数（Rollout）は16回と128回の2種類で敵対学習を行った。

データセット

研究に用いる文データは青空文庫から収集した[12)]。

GANの訓練データには、夏目漱石の著書から『坊っちゃん』『門』『夢十夜』『私の個人主義』の4,504文を用いた。その他の識別タスクの学習データには、GANの訓練データに加え、太宰治、および、芥川龍之介の著書から『人間失格』『グッド・バイ』『走れメロス』『畜犬談』『女生徒』（4,092文）『鼻』『羅生門』『戯作三昧』『地獄変』『河童』『或阿呆の一生』『歯車』『藪の中』（2,532文）を加え、夏目漱石の文には1、その他の文には0の正解ラベルを付した。

評価データは、夏目漱石の著書から『吾輩は猫である』『心』『それから』の文の中からランダムで抽出した3,000文、太宰治の著書から『トカトントン』『お伽草子』『あさましきもの』『HUMAN LOST』の全2,114文、芥川龍之介の著書から『トロッコ』『蜘蛛の糸』『杜子春』『芋粥』『犬と笛』『煙草と悪魔』『舞踏会』『猿蟹合戦』『仙人』『大川の水』『南京の基督』の全847文を用い、夏目漱石の文か、そうでないかの識別を行った。なお、訓練データにおいて夏目漱石の文をランダム抽出しているのは、太宰治と芥川龍之介の文数を加えた数に合わせるためである。

評価指標

本研究では、識別を2値分類問題、すなわち、夏目漱石の文か、そうでないか、を扱っているので、評価指標には、精度（Accuracy）、適合率（Precision）、再現率（Recall）、F値（F1 Score）を用いた。以下では、それぞれについて簡単に説明を加える。

精度は全データの内、正しく予測した割合を測定した指標で、混同行列の要素を使用し表す。ここで、混同行列とは分類したデータの正解・不正解の件数をまとめた表であり、2値分類の場合、4つに区分分けされる（Fig. 7）。

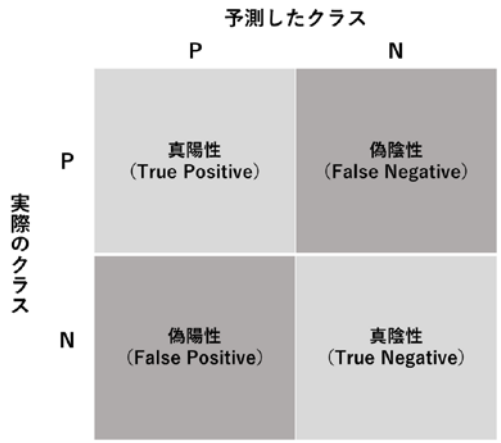

Fig. 7　混同行列（２値分類の場合）

４つの区分はそれぞれ真陽性（True Positive）、真陰性（True Negative）、偽陽性（False Positive）、偽陰性（False Negative）と呼ばれる。真（True）と偽（False）は予測したクラスと実際のクラスが一致したか否かを表し、陽性（Positive）と陰性（Negative）は予測したクラスが何かを表す。すなわち、真陽性と真陰性は予測結果が正解、偽陽性と偽陰性は予測結果が不正解であることということである。

正解率はこれらを用いて全データ中の正解データの割合を以下の式で算出する。

$$\text{正解率} = \frac{\text{真陽性} + \text{真陰性}}{\text{真陽性} + \text{偽陽性} + \text{真陰性} + \text{偽陰性}}$$

次に、再現率はデータ内の陽性数に対して正しく予測した割合、適合率は陽性と予測した数の中で正しく予測した割合を、それぞれ、表す。具体的に、再現率と適合率は以下の式で算出される。

$$\text{再現率} = \frac{\text{真陽性}}{\text{真陽性} + \text{偽陰性}}$$

$$\text{適合率} = \frac{\text{真陽性}}{\text{真陽性} + \text{偽陽性}}$$

最後に、F 値は再現率と適合率の調和平均を取ったもので、再現率と適合率のどちらか一方に偏らせずに評価を行なう際に使用され、以下の式で算出する。

$$\text{F1} = \frac{2 \cdot \text{適合率} \cdot \text{再現率}}{\text{適合率} + \text{再現率}}$$

本研究においては、混同行列、および、これら４つの評価指標によって得られた識別機の性能を評価する。

結果

以下では、研究方法のデータセットの節で説明した学習用データセットで学習を行い、その後、評価用のデータセットを用いて夏目漱石の文か、そうでないかを識別した結果を述べる。上述の通り、教師あり学習を行っている LSTM 識別器、CNN 識別器、TextCNN 識別器においては、夏目漱石の著書（4 冊）に含まれている文に 1 のラベルを付け、その他の太宰治、および、芥川龍之介の著書（合計13冊）に含まれた文には 0 のラベルを付けた後に学習を行っている。一方、識別機の構築に GAN を用いる際には、教師あり学習で用いたものと同じ夏目漱石の著書（4 冊）に含まれている文のみを訓練データに用いている。なお、評価データについては、 4 つのモデルとも同じものを用い、混同行列の配置は Fig. 8 の通りである。

	予測したクラス	
実際のクラス	夏目漱石を正しく識別（True Positive）	夏目漱石をその他と識別（False Negative）
	その他を夏目漱石と識別（False Positive）	その他を正しく識別（True Negative）

Fig. 8　混同行列の内容

学習済み LSTM 識別器の評価データに対する識別結果の混同行列を Table 1 に示す。この時の正解率は0.732、適合率が0.669、再現率が0.911、F 値が0.771である。なお、学習データに対しては、正解率で0.998であった。

Table 1　評価データに対する LSTM 識別器の混同行列

	予測したクラス	
実際のクラス	1666	1334
	265	2696

学習済み CNN 識別器の評価データに対する識別結果の混同行列を Table 2 に示す。この時の正解率は0.740、適合率が0.831、再現率が0.607、F 値が0.702である。なお、学習データに対しては、正解率で0.997であった。

Table 2　評価データに対する CNN 識別器の混同行列

	予測したクラス	
実際のクラス	1821	1179
	370	2591

学習済み TextCNN 識別器の評価データに対する識別結果の混同行列 Table 3 に示す。この時の正解率は0.735、適合率が0.683、再現率が0.870、F 値が0.765である。なお、学習データに対しては、正解率で0.997であった。

Table 3　評価データに対する TextCNN 識別器の混同行列

	予測したクラス	
実際のクラス	1805	1195
	385	2576

SeqGAN によって学習させた TextCNN 識別器の評価データに対する識別結果の混同行列を Table 4

に示す。この時の正解率は0.494、適合率が0.497、再現率が0.407、F 値が0.447である。なお、敵対学習終了時において、生成文と訓練の文との識別の正解率は1.000であった。

Table 4　評価データに対する敵対学習によって構築した TextCNN 識別器の混同行列

	予測したクラス	
実際のクラス	1220	1780
	1234	1727

以上、4 つの識別機における指標をまとめると Table 5 となる。

Table 5　評価データに対する各識別器の評価指標のまとめ

	LSTM	CNN	TextCNN	TextCNN（GAN）
正解率	0.732	0.74	0.735	0.494
適合率	0.669	0.831	0.638	0.497
再現率	0.911	0.607	0.87	0.407
F 値	0.771	0.702	0.702	0.447

考察

Table 5 から各識別器の特徴として、LSTM 識別器は再現性が高く、CNN 識別器は適合率が高いことが見て取れる。すなわち、前者はその他も夏目漱石の文と認識してしまうこともあるが、夏目漱石の文は夏目漱石の文と識別することに長けている、一方、後者は夏目漱石の文を見逃してしまうことはあるが、夏目漱石の文と識別した際の的中率は高いという特徴を持っている。

また、混同行列からは教師あり学習で構築した 3 つのモデルに共通した特徴として、False Positive の数が False Negative と比較して少ないことが見て取れる。つまり、その他の作家の文を夏目漱石の文と識別することは少ないが、夏目漱石の文をその他に識別することは多いということである。この特徴が、本研究の作家の組み合わせに特有のものなのか、一般的に、1 文の識別タスクにおいて教師あり学習を採用した場合に、Negative を Positive と識別することは少なく、Positive を Negative と識別することが多いのかは本研究のみでは結論を得られないので、さらなる実験、検討を要する。

そして、1 文のみによって、夏目漱石の文か否かを識別するタスクにおいて、教師あり学習によって得られた識別器の正解率は0.73～0.74程度であったのに対して、GAN によって訓練された識別器の正解率は0.494と著しく低いものとなった。ただし、教師あり学習で得られた全く同じネットワーク構造を有する TextCNN の正解率が0.735であるため、TextCNN 構造自体のポテンシャルは、最低でも、正解率で0.7程度はあるため、学習において、何らかの課題があることが推察される。

この点について、敵対学習時において、学習回数が200回目程度から、生成文と訓練の文との識別の正解率が1.000になってしまったのが一つの要因と考えられる。GAN における敵対学習の大きな困難さの 1 つには、Generator と Discriminator の学習速度の調整がある。すなわち、Generator よりも Discriminator の学習が極端に速く進んでしまい、その結果、完璧に生成データと訓練データを識別できるようになってしまうと、数式的には、(1) 式が最大値である 0 から変わらなくなってしまい、Generator において何をしても結果が変わらないので学習の方向性が定まらず、学習が進まなくなってしまう。本研究の仮説を言葉にすると『敵対学習によって“夏目漱石らしい”特徴を反映した文を生成する Generator と Discriminator が敵対的に競うことによって、Discriminator が“夏目漱石らしい”かどうかを識別できるようになる』となるが、Generator が“夏目漱石らしい”文を生成できるようになっていな

いので、Discriminator が十分に学習できていない可能性があるということである。

そこで、Generator の学習が Discriminator の学習よりも相対的に速く進むよう、Rollout の回数を128回に増やし、他の条件は変えず、敵対学習を行ったところ、評価データに対する混同行列は Table 6 となり、正解率は0.530、適合率が0.561、再現率が0.0.31、F 値が0.339 となり、正解率がやや改善した。なお、敵対学習終了時において、生成文と訓練の文との識別の正解率は0.973であった。

得られた混同行列で特徴的なのが、評価データに対する識別の出力が大きくその他に傾いた点である。すなわち、識別機としては、夏目漱石らしい文章は評価データに含まれた文の中には、少ないと判断しているということである。

Table 6　評価データに対する敵対学習によって構築した TextCNN 識別器の混同行列

	予測したクラス	
実際のクラス	930	2070
	729	2232

教師あり学習においては、学習時に夏目漱石の文もその他 2 人の作家の文も入力されるので、Positive と識別すべき文も Negative と識別すべき文の特徴も識別器は学習しているのに対し、敵対学習の場合は、Positive と識別すべきデータの特徴しか学習で扱わない。したがって、後者の識別性能を向上させるためには、Generator が生成する文の“質”が重要であることが本研究でわかった。実際、Rollout の数を増加させることで、正解率が向上している。今後、Generator の学習を工夫し、テーマに沿った文の生成を向上させることで、Discriminator の識別率も向上すると考えられる。その方針には、訓練データ数の増加、学習の最適化、学習回数の増加、GAN の種類の変更などが考えられる。

まとめ

本研究では、GAN をベースとした小論文・レポートの評価システムの構築を目標に、敵対学習によって得られた Discriminator による 1 文の識別が可能かを検討した。その結果、現時点で、利用可能といえる識別器の取得には至らなかったが、敵対学習を行う際、Generator の学習を検討することで、性能向上が可能となることが示唆された。

謝辞

本研究は JSPS 科研費18K13240の助成を受けて実施されたものです。

参考文献

1）I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “*Generative adversarial nets*”, Advances in Neural Information Processing Sys tems, pp. 2672-2680, 2014.

2）Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “*Gradient-based learning applied to document recognition*”, Proc. of the IEEE, pages 2278–2324, 1998.

3）岡谷貴之，『深層学習』．講談社，2015.

4）Y.Kim, *“Convolutional Neural Networks for Sentence Classification”*, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.

5）石岡恒憲，人工知能学会誌，Vol.23，pp.17-24，2008.

6）岩田具治『トピックモデル』講談社 2015年.

7）T. Mikolov, K. Chen, G. Corrado and J. Dean, “Efficient Estimation of Word Representations in Vector Space”. https://arxiv.org/abs/1301.3781
8）斎藤康毅,『ゼロから作る Deep Learning 2』オライリー・ジャパン．2018.
9）S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory”, Journal Neural Computation, vol 9, pages 1735-1780, 1997.
10）岩田一樹,『敵対生成ネットワークによる文書生成』，感性福祉研究所年報，21，pp.53-64，2020.
11）L. Yu, W. Zhang, J.Wang, and Y. Yu, “SeqGAN: Sequence generative adversarial nets with policy gradient”, AAAI, pp.2852-2858, 2017.
12）青空文庫 https://www.aozora.gr.jp/